A 2019 Guide to Speech Synthesis with Deep Learning

Derrick Mwiti

Aug 28 · 13 min read

Photo by Daniel Sandvik on Unsplash

Artificial production of human speech is known as speech synthesis. This machine learning-based technique is applicable in text-to-speech, music generation, speech generation, speech-enabled devices, navigation systems, and accessibility for visually-impaired people.

In this article, we'll look at research and model architectures that have been written and developed to do just that using deep learning.

But before we jump in, there are a couple of specific, traditional strategies for speech synthesis that we need to briefly outline: **concatenative** and **parametric**.

In the concatenative approach, speeches from a large database are used to generate new, audible speech. In a case where a different style of speech is needed, a new database of audio voices is used. This limits the scalability of this approach.

The parametric approach uses a recorded human voice and a function with a set of parameters that can be modified to change the voice.

These two approaches represent the old way of doing speech synthesis. Now let's look at the new ways of doing it using deep learning. Here's the research we'll cover in order to examine popular and current approaches to speech synthesis:

- WaveNet: A Generative Model for Raw Audio
- Tacotron: Towards End-toEnd Speech Synthesis
- Deep Voice 1: Real-time Neural Text-to-Speech
- Deep Voice 2: Multi-Speaker Neural Text-to-Speech
- Deep Voice 3: Scaling Text-to-speech With Convolutional Sequence Learning
- Parallel WaveNet: Fast High-Fidelity Speech Synthesis
- Neural Voice Cloning with a Few Samples
- VoiceLoop: Voice Fitting and Synthesis via A Phonological Loop
- Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions

. . .

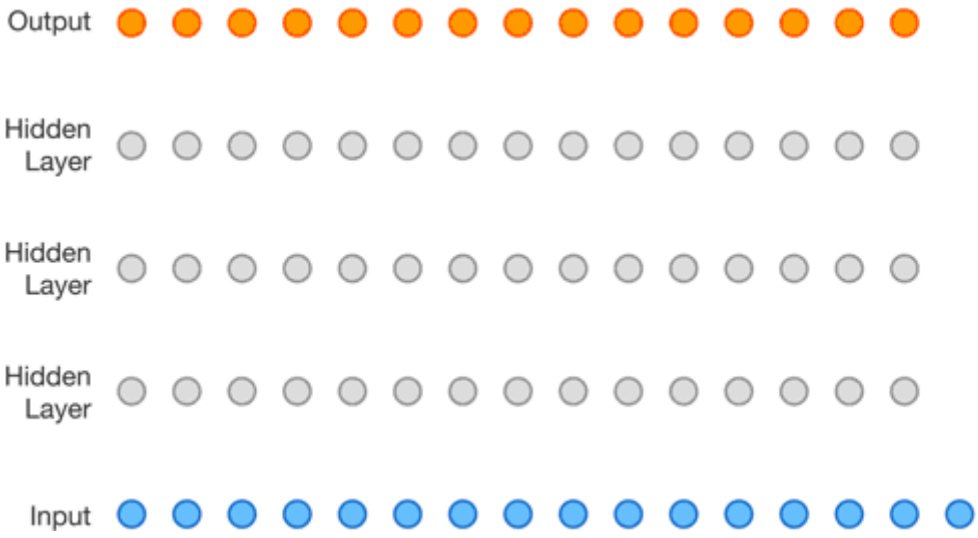
WaveNet: A Generative Model for Raw Audio

The authors of this paper are from Google. They present a neural network for generating raw audio waves. Their model is fully probabilistic and autoregressive, and it generates state-of-the-art text-to-speech results for both English and Mandarin.

WaveNet: A Generative Model for Raw Audio

This paper introduces WaveNet, a deep neural network for generating raw audio waveforms. The model is fully...

arxiv.org



source

WaveNet is an audio generative model based on the PixelCNN. It's capable of producing audio that's very similar to a human voice.

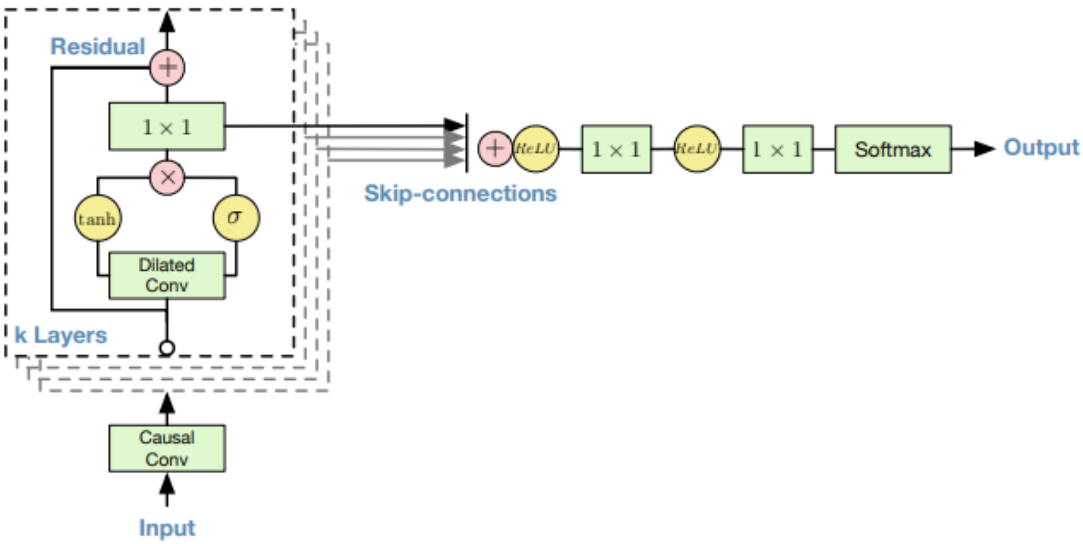


Figure 4: Overview of the residual block and the entire architecture.

source

Google Wavenet | Google I/O 2018

In this generative model, each audio sample is conditioned on the previous audio sample. The conditional probability is modeled by a stack of convolutional layers. This network doesn't have pooling layers, and the output of the model has the same time dimensionality as the input.

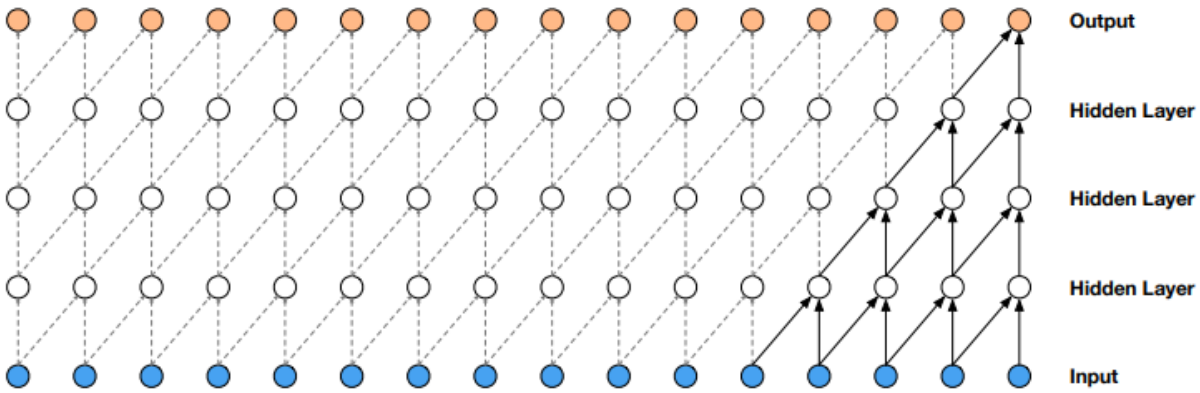


Figure 2: Visualization of a stack of causal convolutional layers.

source

The use of casual convolutions in the architecture ensures that the model doesn't violate the ordering of how the data is modeled. In this model, each predicted voice sample is fed back to the network to aid in predicting the next one. Since casual convolutions don't have a recurrent connection, they're faster to train than RNNs.

One of the major challenges of using casual convolutions is that they require many layers in order to increase the receptive field. To solve this challenge, the authors use

dilated convolutions. Dilated convolutions enable networks to have a large receptive field but with a few layers. Modeling the conditional distributions over the individual audio samples is done using a softmax distribution.

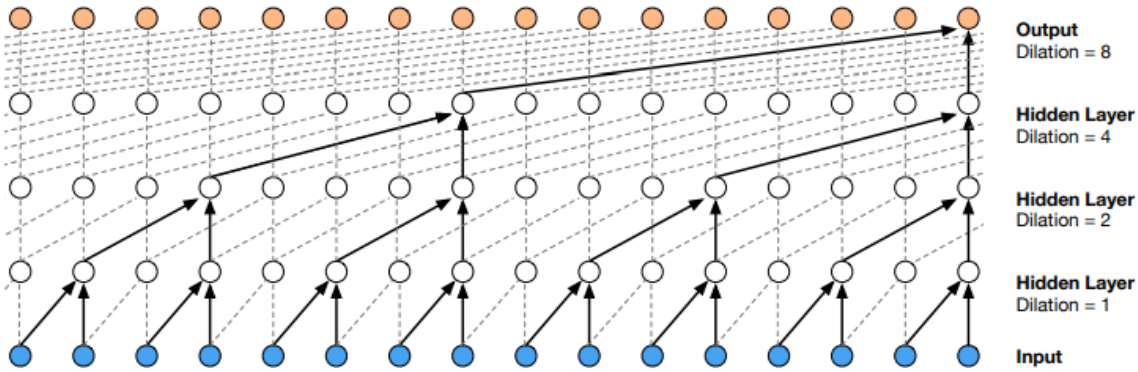


Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

source

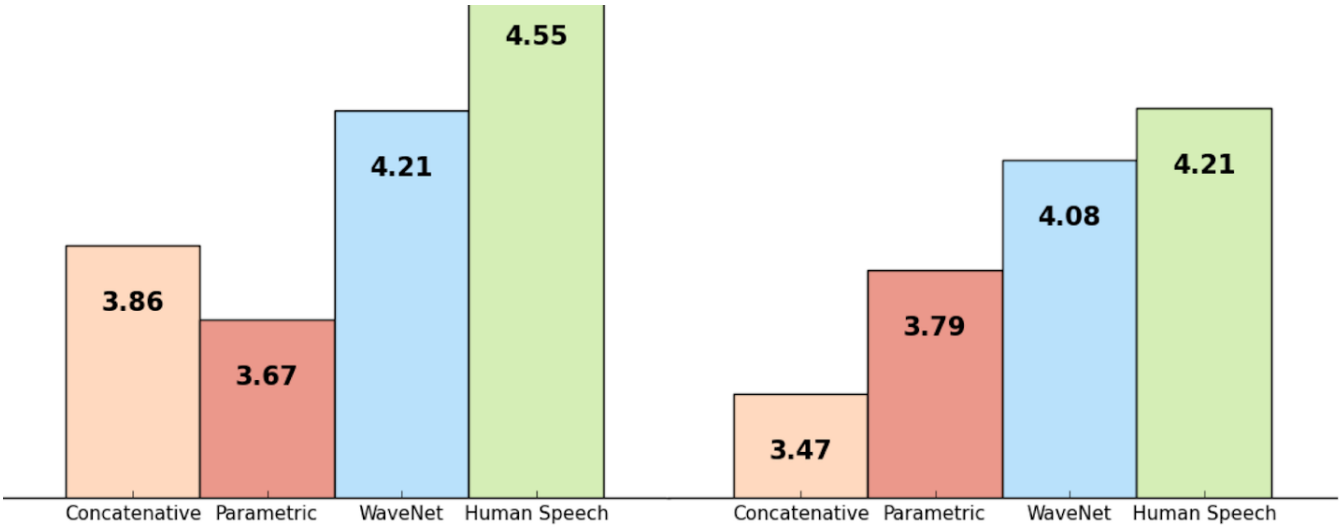
The model is evaluated on multispeaker speech generation, text-to-speech, and music audio modeling. The MOS (Mean Opinion Score) is used for this testing. It measures the quality of voice. It's basically the opinion of a person about the voice quality. It is a number between one and five, with five being the best quality.

	Subjective 5-scale MOS in naturalness	
Speech samples	North American English	Mandarin Chinese
LSTM-RNN parametric	3.67 ± 0.098	3.79 ± 0.084
HMM-driven concatenative	3.86 ± 0.137	3.47 ± 0.108
WaveNet (L+F)	$\mathbf{4.21} \pm 0.081$	$\mathbf{4.08} \pm 0.085$
Natural (8-bit μ-law)	4.46 ± 0.067	4.25 ± 0.082
Natural (16-bit linear PCM)	4.55 ± 0.075	4.21 ± 0.071

Table 1: Subjective 5-scale mean opinion scores of speech samples from LSTM-RNN-based statistical parametric, HMM-driven unit selection concatenative, and proposed WaveNet-based speech synthesizers, 8-bit μ-law encoded natural speech, and 16-bit linear pulse-code modulation (PCM) natural speech. WaveNet improved the previous state of the art significantly, reducing the gap between natural speech and best previous model by more than 50%.

source

The figure below shows the quality of waveNets on a scale of 1–5.



source

. . .

> The latest in deep learning — from a source you can trust. Sign up for a weekly dive into all things deep learning, curated by experts working in the field.

. . .

Tacotron: Towards End-toEnd Speech Synthesis

The authors of this paper are from Google. Tacotron is an end-to-end generative text-to-speech model that synthesizes speech directly from text and audio pairs. Tacotron achieves a 3.82 mean opinion score on US English. Tacotron generates speech at frame-level and is, therefore, faster than sample-level autoregressive methods.

Tacotron: Towards End-to-End Speech Synthesis

A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic...

arxiv.org

The model is trained on audio and text pairs, which makes it very adaptable to new datasets. Tacotron has a seq2seq model that includes an encoder, an attention-based decoder, and a post-processing net. As seen in the architecture diagram below, the

model takes characters as input and outputs a raw spectrogram. This spectrogram is then converted to waveforms.

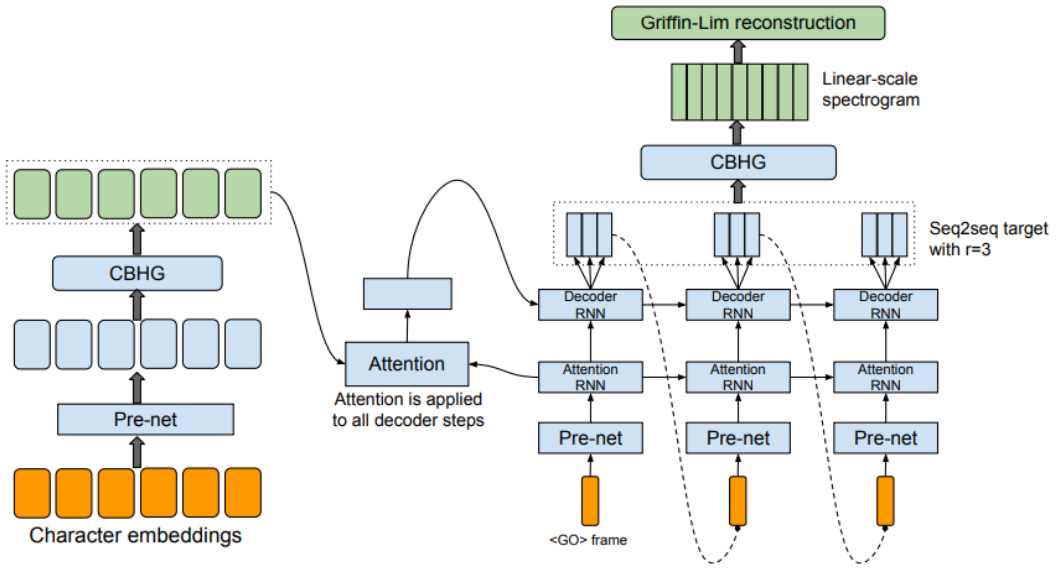


Figure 1: *Model architecture. The model takes characters as input and outputs the corresponding raw spectrogram, which is then fed to the Griffin-Lim reconstruction algorithm to synthesize speech.*

source

The figure below shows what the CBHG module looks like. It consists of 1-D convolution filters, highway networks, and a bidirectional GRU (Gated Recurrent Unit).

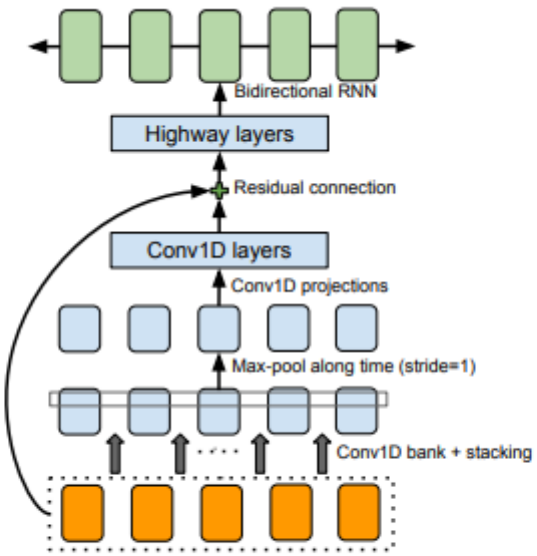


Figure 2: The CBHG (1-D convolution bank + highway network + bidirectional GRU) module adapted from Lee et al. (2016).

source

A character sequence is fed to the encoder, which extracts sequential representations of text. Each character is represented as a one-hot vector and embedded into a continuous vector. Non-linear transformations are then added, followed by a dropout layer to reduce overfitting. This, in essence, reduces the mispronunciation of words.

The decode used is a *tanh* content-based attention decoder. The waveforms are then generated using the Griffin-Lim algorithm. The hyper-parameters used for this model are shown below.

Table 1: Hyper-parameters and network architectures. "conv-k-c-ReLU" denotes 1-D convolution with width k and c output channels with ReLU activation. FC stands for fully-connected.

Spectral analysis	*pre-emphasis*: 0.97; *frame length*: 50 ms; *frame shift*: 12.5 ms; *window type*: Hann
Character embedding	256-D
Encoder CBHG	*Conv1D bank*: K=16, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-128-ReLU $\rightarrow$ conv-3-128-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Encoder pre-net	FC-256-ReLU $\rightarrow$ Dropout(0.5) $\rightarrow$ FC-128-ReLU $\rightarrow$ Dropout(0.5)
Decoder pre-net	FC-256-ReLU $\rightarrow$ Dropout(0.5)$\rightarrow$ FC-128-ReLU $\rightarrow$ Dropout(0.5)
Decoder RNN	2-layer residual GRU (256 cells)
Attention RNN	1-layer GRU (256 cells)
Post-processing net CBHG	*Conv1D bank*: K=8, conv-k-128-ReLU *Max pooling*: stride=1, width=2 *Conv1D projections*: conv-3-256-ReLU $\rightarrow$ conv-3-80-Linear *Highway net*: 4 layers of FC-128-ReLU *Bidirectional GRU*: 128 cells
Reduction factor (r)	2

source

The figure below shows the performance of Tacotron compared to other alternatives.

Table 2: 5-scale mean opinion score evaluation.

	mean opinion score
Tacotron	3.82 ± 0.085
Parametric	3.69 ± 0.109
Concatenative	4.09 ± 0.119

source

. . .

Deep Voice 1: Real-time Neural Text-to-Speech

The authors of this paper are from Baidu's Silicon Valley Artificial Intelligence Lab. Deep Voice is a text-to-speech system developed using deep neural networks.

> **Deep Voice: Real-time Neural Text-to-Speech**
>
> We present Deep Voice, a production-quality text-to-speech system constructed entirely from deep neural networks. Deep...
>
> arxiv.org

It has five major building blocks:

- A segmentation model for locating phoneme boundaries with deep neural networks using connectionist temporal classification (CTC) loss.
- A grapheme-to-phoneme conversion model (grapheme-to-phoneme is the process of using rules to generate a word's pronunciation).
- A phoneme duration prediction model.
- A fundamental frequency prediction model.
- An audio synthesis model using a variant of WaveNet that uses fewer parameters.

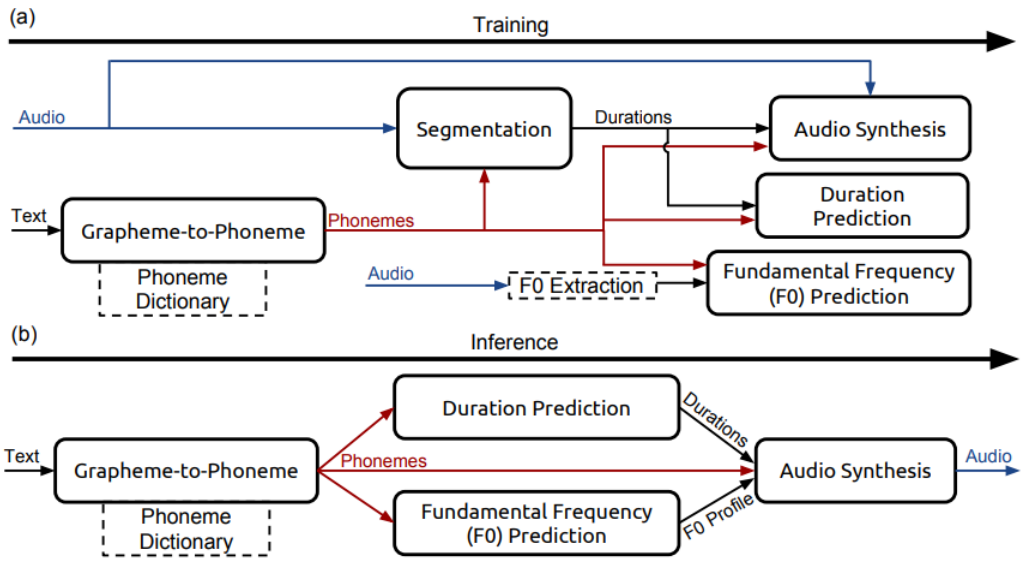


Figure 1. System diagram depicting (a) training procedure and (b) inference procedure, with inputs on the left and outputs on the right. In our system, the duration prediction model and the F0 prediction model are performed by a single neural network trained with a joint loss. The grapheme-to-phoneme model is used as a fallback for words that are not present in a phoneme dictionary, such as CMUDict. Dotted lines denote non-learned components.

source

The grapheme-to-phoneme model converts English characters to phonemes. The segmentation model identifies where each phoneme begins and ends in an audio file. The phoneme duration model predicts the duration of every phoneme in a phoneme sequence.

The fundamental frequency model predicts whether a phoneme is voiced. The audio synthesis model synthesizes audio by combining the output of the grapheme-to-phoneme, phoneme duration, and fundamental frequency prediction models.

Here's how this model fares compared to other models.

Deep Voice: Real-time Neural TTS

Type	Model Size	MOS±CI
Ground Truth (48 kHz)	None	4.75 ± 0.12
Ground Truth	None	4.45 ± 0.16
Ground Truth (companded and expanded)	None	4.34 ± 0.18
Synthesized	$\ell = 40, r = 64, s = 256$	3.94 ± 0.26
Synthesized (48 kHz)	$\ell = 40, r = 64, s = 256$	3.84 ± 0.24
Synthesized (Synthesized F0)	$\ell = 40, r = 64, s = 256$	2.76 ± 0.31
Synthesized (Synthesized Duration and F0)	$\ell = 40, r = 64, s = 256$	2.00 ± 0.23
Synthesized (2X real-time inference)	$\ell = 20, r = 32, s = 128$	2.74 ± 0.32
Synthesized (1X real-time inference)	$\ell = 20, r = 64, s = 128$	3.35 ± 0.31

Table 1. Mean Opinion Scores (MOS) and 95% confidence intervals (CIs) for utterances. This MOS score is a relative MOS score obtained by showing raters the same utterance across all the model types (which encourages comparative rating and allows the raters to distinguish finer grained differences). Every batch of samples also includes the ground truth 48 kHz recording, which makes all our ratings comparative to natural human voices. 474 ratings were collected for every sample. Unless otherwise mentioned, models used phoneme durations and F0 extracted from the ground truth, rather than synthesized by the duration prediction and frequency prediction models, as well as a 16384 Hz audio sampling rate.

• • •

Deep Voice 2: Multi-Speaker Neural Text-to-Speech

This paper represents the second iteration of Deep Voice by Baidu Silicon Valley Artificial Intelligence Lab. They introduce a method for augmenting neural text-to-speech with low dimensional trainable speaker embeddings to produce various voices from a single model.

The model is based on a similar pipeline as DeepVoice 1. However, it represents a significant improvement in audio quality. The model is able to learn hundreds of unique voices from less than half an hour of data per speaker.

Deep Voice 2: Multi-Speaker Neural Text-to-Speech

We introduce a technique for augmenting neural text-to-speech (TTS) with

low dimensional trainable speaker embeddings to...

arxiv.org

The authors also introduce a WaveNet-based spectrogram-to-audio neural vocoder, which is then used with Tacotron in place of Griffin-Lim audio generation. The main focus of this paper is to handle multiple speakers with fewer data from each speaker. The general architecture is similar to Deep Voice 1. The training process of Deep Voice 2 is depicted in the figure below.

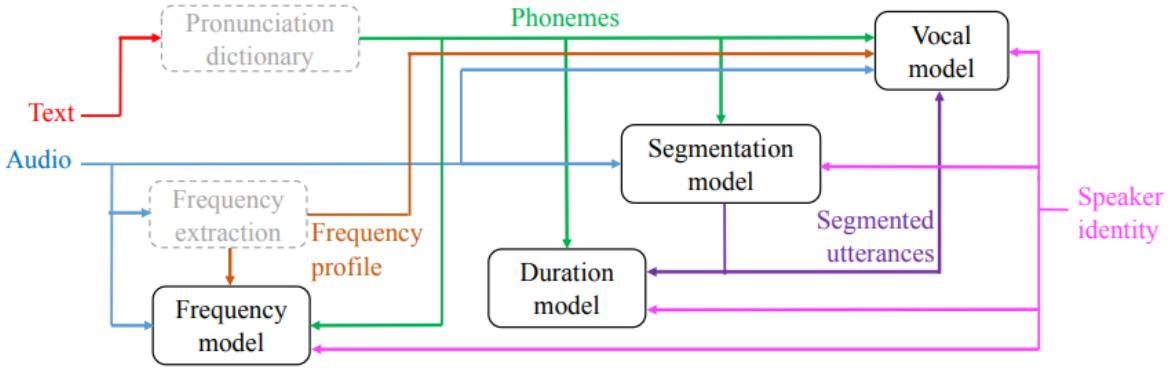


Figure 5: System diagram for training procedure for Deep Voice 2.

source

The major difference between Deep Voice 2 and Deep Voice 1 is the separation of the phoneme duration and frequency models. Deep Voice 1 has a single model for jointly predicting the phoneme duration and frequency profile; in Deep Voice 2, the phoneme durations are predicted first and then they are used as inputs to the frequency model.

The segmentation model in Deep Voice 2 is a convolutional-recurrent architecture with connectionist temporal classification (CTC) loss applied to classify phoneme pairs. The major modification in Deep Voice 2 is the addition of batch normalization and residual connections in the convolutional layers. Its vocal model is based on a WaveNet architecture.

Synthesizing speech from multiple speakers is done by augmenting each model with a single low-dimensional level speaker embedding vector per speaker. Weight sharing between speakers is achieved by storing speaker-dependent parameters in a very low-dimensional vector.

The initial states of the recurrent neural network (RNN) are produced using speaker embeddings. A uniform distribution is used to randomly initialize the speaker

embeddings and trained jointly using backpropagation. Speaker embeddings are incorporated in multiple portions of the model in order to ensure that each speaker's unique voice signature is factored in.

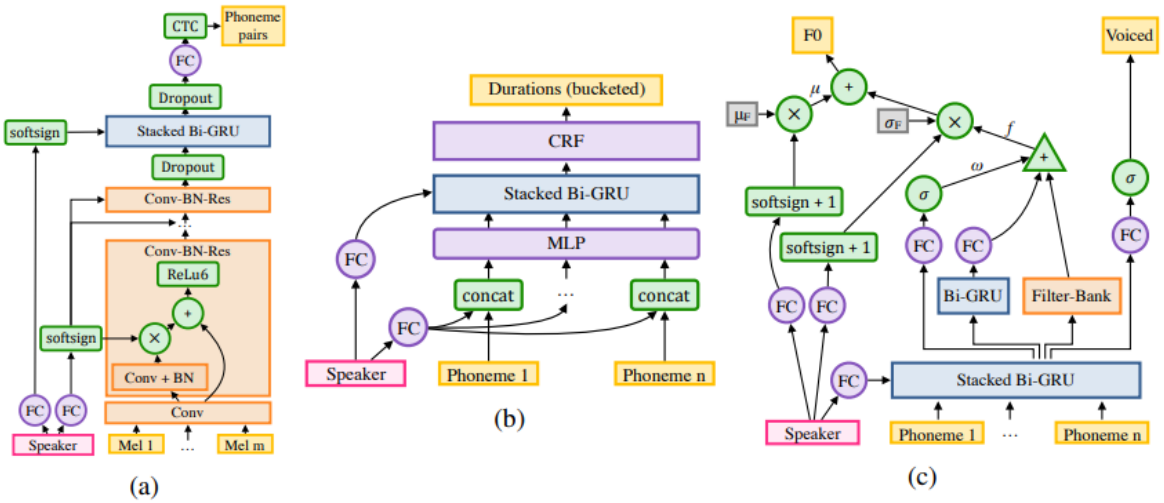


Figure 2: Architecture for the multi-speaker (a) segmentation, (b) duration, and (c) frequency model.

source

Let's now see how this model performs in comparison to other models.

Dataset	Multi-Speaker Model	Samp. Freq.	MOS	Acc.
VCTK	Deep Voice 2 (20-layer WaveNet)	16 KHz	2.87±0.13	99.9%
VCTK	Deep Voice 2 (40-layer WaveNet)	16 KHz	3.21±0.13	100 %
VCTK	Deep Voice 2 (60-layer WaveNet)	16 KHz	3.42±0.12	99.7%
VCTK	Deep Voice 2 (80-layer WaveNet)	16 KHz	3.53±0.12	99.9%
VCTK	Tacotron (Griffin-Lim)	24 KHz	1.68±0.12	99.4%
VCTK	Tacotron (20-layer WaveNet)	24 KHz	2.51±0.13	60.9%
VCTK	Ground Truth Data	48 KHz	4.65±0.06	99.7%
Audiobooks	Deep Voice 2 (80-layer WaveNet)	16 KHz	2.97±0.17	97.4%
Audiobooks	Tacotron (Griffin-Lim)	24 KHz	1.73±0.22	93.9%
Audiobooks	Tacotron (20-layer WaveNet)	24 KHz	2.11±0.20	66.5%
Audiobooks	Ground Truth Data	44.1 KHz	4.63±0.04	98.8%

Table 2: MOS and classification accuracy for all multi-speaker models. To obtain MOS, we use crowdMOS toolkit as detailed in Table 1. We also present classification accuracies of the speaker discriminative models (see Appendix E for details) on the samples, showing that the synthesized voices are as distinguishable as ground truth audio.

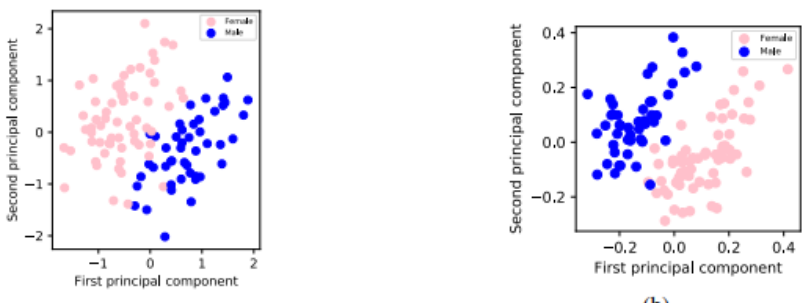


Figure 4: Principal components of the learned speaker embeddings for the (a) 80-layer vocal model and (b) character-to-spectrogram model for VCTK dataset. See Appendix D.3 for details.

Model	Samp. Freq.	MOS
Deep Voice 1	16 KHz	2.05 ± 0.24
Deep Voice 2	16 KHz	2.96 ± 0.38
Tacotron (Griffin-Lim)	24 KHz	2.57 ± 0.28
Tacotron (WaveNet)	24 KHz	4.17 ± 0.18

Table 1: Mean Opinion Score (MOS) evaluations with 95% confidence intervals of Deep Voice 1, Deep Voice 2, and Tacotron. Using the crowdMOS toolkit, batches of samples from these models were presented to raters on Mechanical Turk. Since batches contained samples from all models, the experiment naturally induces a comparison between the models.

source

. . .

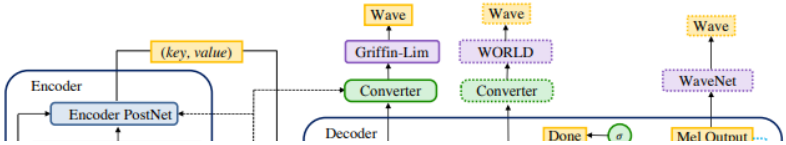
Deep Voice 3: Scaling Text-to-speech With Convolutional Sequence Learning

In the third iteration of Deep Voice, the authors introduce is a fully-convolutional attention-based neural text-to-speech (TTS) system.

> **Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning**
>
> We present Deep Voice 3, a fully-convolutional attention-based neural text-to-speech (TTS) system. Deep Voice 3 matches...
>
> arxiv.org

The authors propose a fully-convolutional character-to-spectrogram architecture that enables fully parallel computation. The architecture is an attention-based sequence-to-sequence model. The model was trained on the LibriSpeech ASR dataset.

The proposed architecture is able to convert textual features such as characters, phonemes, and stresses into different vocoder parameters. Some of these include mel-band spectrograms, linear-scale log magnitude spectrograms, fundamental frequency, spectral envelope, and aperiodicity parameters. These vocoder parameters are then used as the input for the audio waveform synthesis model.



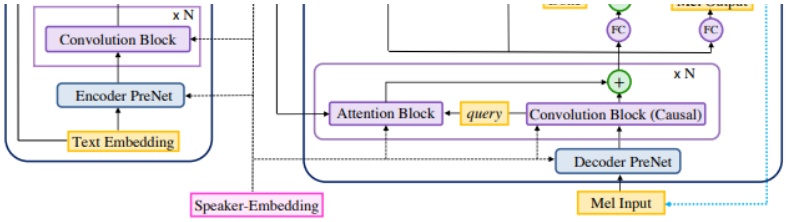


Figure 1: Deep Voice 3 uses residual convolutional layers to encode text into per-timestep *key* and *value* vectors for an attention-based decoder. The decoder uses these to predict the mel-scale log magnitude spectrograms that correspond to the output audio. (Light blue dotted arrows depict the autoregressive process during inference.) The hidden states of the decoder are then fed to a converter network to predict the vocoder parameters for waveform synthesis. See Appendix A for more details.

source

The architecture is composed of the following:

- Encoder — a fully-convolutional encoder that converts textual features to an internal learned representation.
- Decoder — a fully-convolutional causal decoder that decodes the learned representations in an autoregressive manner.
- Converter — a fully-convolutional post-processing network that predicts the final vocoder parameters.

For text pre-processing, the authors' uppercase text input characters, remove punctuation marks, end each utterance with a period or question mark, and replace spaces with a special character that indicates the length of a pause.

The figure below is a comparison of the performance of this model with other alternative models.

Model	Mean Opinion Score (MOS)
Deep Voice 3 (Griffin-Lim)	3.62 ± 0.31
Deep Voice 3 (WORLD)	3.63 ± 0.27
Deep Voice 3 (WaveNet)	3.78 ± 0.30
Tacotron (WaveNet)	3.78 ± 0.34
Deep Voice 2 (WaveNet)	2.74 ± 0.35

Table 2: Mean Opinion Score (MOS) ratings with 95% confidence intervals using different waveform synthesis methods. We use the crowdMOS toolkit (Ribeiro et al., 2011); batches of samples from these models were presented to raters on Mechanical Turk. Since batches contained samples from all models, the experiment naturally induces a comparison between the models.

Model	MOS (VCTK)	MOS (LibriSpeech)
Deep Voice 3 (Griffin-Lim)	3.01 ± 0.29	2.37 ± 0.24
Deep Voice 3 (WORLD)	3.44 ± 0.32	2.89 ± 0.38
Deep Voice 2 (WaveNet)	3.69 ± 0.23	-
Tacotron (Griffin-Lim)	2.07 ± 0.31	-
Ground truth	4.69 ± 0.04	4.51 ± 0.18

Table 3: MOS ratings with 95% confidence intervals for audio clips from neural TTS systems on multi-speaker datasets. We also use crowdMOS toolkit; batches of samples including ground truth were presented to human raters. Multi-speaker Tacotron implementation and hyperparameters are

based on Arık et al. (2017), which is a proof-of-concept implementation. Deep Voice 2 and Tacotron systems were not trained for the LibriSpeech dataset due to prohibitively long time required to optimize hyperparameters.

source

. . .

. . .

Parallel WaveNet: Fast High-Fidelity Speech Synthesis

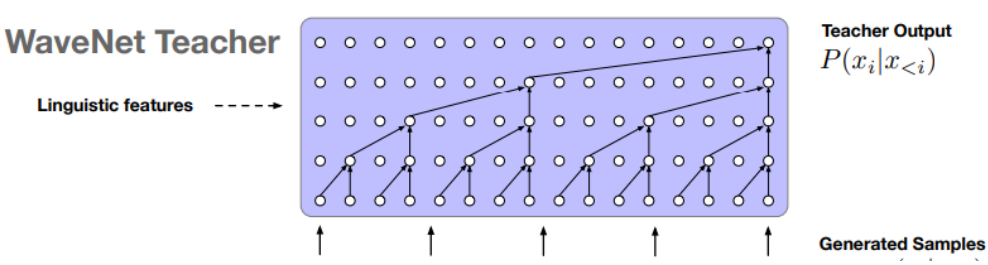
The authors of this paper are from Google. They introduce a method known as *Probability Density Distillation,* which trains a parallel feed-forward network from a trained WaveNet. The method has been built by marrying the best features of Inverse autoregressive flows (IAFs) and WaveNet. These features represent the efficient training of WaveNet and the efficient sampling of IAF networks.

Parallel WaveNet: Fast High-Fidelity Speech Synthesis

The recently-developed WaveNet architecture is the current state of the art in realistic speech synthesis, consistently...

arxiv.org

For training, the authors use a trained WaveNet as a 'teacher', and the parallel WaveNet 'student' learns from this. The goal here is to have the student match the probability of its own samples under the distribution learned from the teacher.



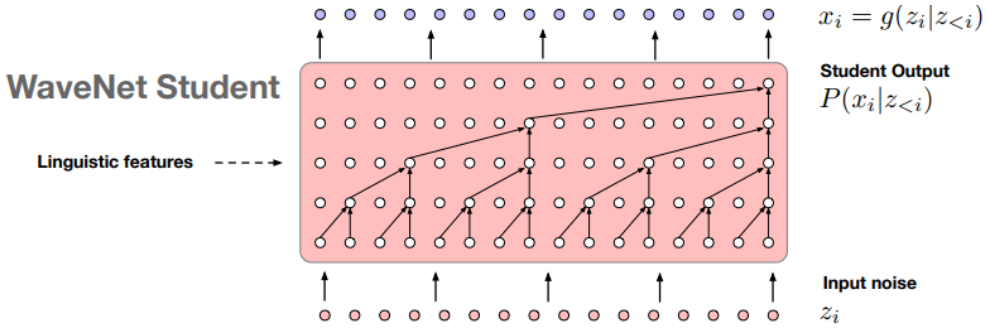


Figure 2: **Overview of Probability Density Distillation**. A pre-trained WaveNet teacher is used to score the samples $\boldsymbol{x}$ output by the student. The student is trained to minimise the KL-divergence between its distribution and that of the teacher by maximising the log-likelihood of its samples under the teacher and maximising its own entropy at the same time.

source

The authors also propose additional loss functions for guiding the student in generating high-quality audio streams:

- Power loss — to ensure that the power in different frequency bands of the speeches is used, as in human speech.
- Perceptual loss — for this loss, the authors experimented with feature reconstruction loss (the Euclidean distance between feature maps in the classifier) and style loss (the Euclidean distance between the Gram matrices). They found that style loss produced better results.
- Contrastive loss that penalizes waveforms that have high likelihood regardless of the conditioning vector.

The figure below shows the performance of this model.

	Parametric	**Concatenative**	**Distilled WaveNet**
English speaker 1 (female - 65h data)	3.88	4.19	4.41
English speaker 2 (male - 21h data)	3.96	4.09	4.34
English speaker 3 (male - 10h data)	3.77	3.65	4.47
English speaker 4 (female - 9h data)	3.42	3.40	3.97
Japanese speaker (female - 28h data)	4.07	3.47	4.23

Table 2: Comparison of MOS scores on English and Japanese with multi-speaker distilled WaveNets. Note that some speakers sounded less appealing to people and always get lower MOS, however distilled parallel WaveNet always achieved significantly better results.

Method	Subjective 5-scale MOS
16kHz, 8-bit μ-law, 25h data:	
LSTM-RNN parametric [27]	3.67 ± 0.098
HMM-driven concatenative [27]	3.86 ± 0.137
WaveNet [27]	4.21 ± 0.081
24kHz, 16-bit linear PCM, 65h data:	
HMM-driven concatenative	4.19 ± 0.097
Autoregressive WaveNet	4.41 ± 0.069
Distilled WaveNet	4.41 ± 0.078

Table 1: Comparison of WaveNet distillation with the autoregressive teacher WaveNet, unit-selection (concatenative), and previous results from [27]. MOS stands for Mean Opinion Score.

source

. . .

Neural Voice Cloning with a Few Samples

The authors of this paper are from Baidu Research. They introduce a neural voice cloning system that learns to synthesize a person's voice from a few audio samples.

The two approaches used are speaker adaptation and speaker encoding. Speaker adaptation works by fine-tuning a multi-speaker generative model, while speaker encoding works by training a separate model to directly infer a new speaker embedding that's applied to the multi-speaker generative model.

Neural Voice Cloning with a Few Samples

Voice cloning is a highly desired feature for personalized speech interfaces. Neural network based speech synthesis has...

arxiv.org

This paper uses Deep Voice 3 as the baseline for the multi-speaker model. For voice cloning, the authors extract speaker characteristics from a speaker and generate audio provided that text from a given speaker is available.

The performance metrics used for the generated audio are speech naturalness and speaker similarity. They propose a speaker encoding method that directly estimates a speaker's embeddings from the audio samples of an unseen speaker.

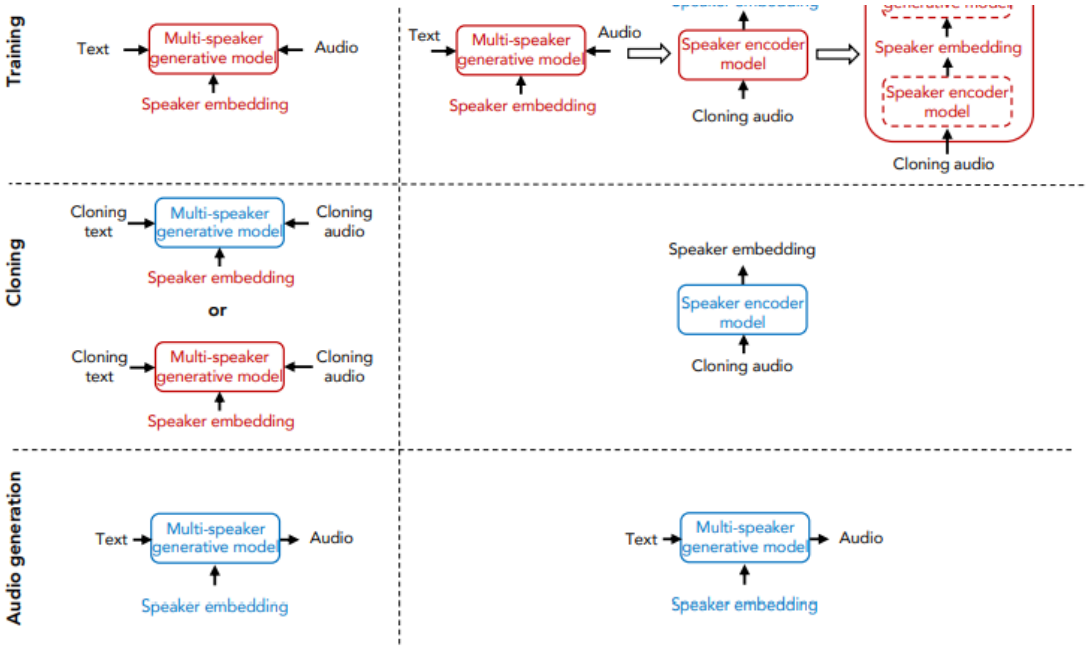


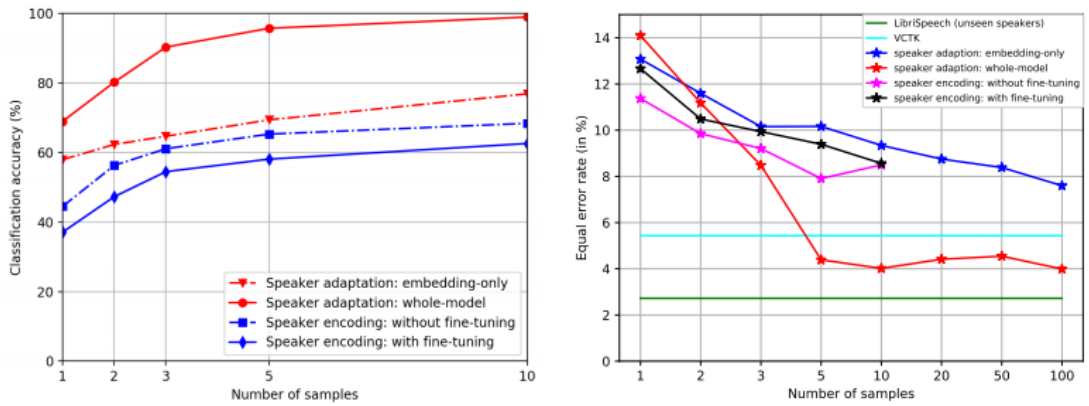
Figure 1: Illustration of speaker adaptation and speaker encoding approaches for voice cloning.

source

Below is a look at how voice cloning performs.

	Speaker adaptation		Speaker encoding	
Approaches	Embedding-only	Whole-model	Without fine-tuning	With fine-tuning
Data	Text and audio		Audio	
Cloning time	~ 8 hours	$\sim 0.5 - 5$ mins	$\sim 1.5 - 3.5$ secs	$\sim 1.5 - 3.5$ secs
Inference time	$\sim 0.4 - 0.6$ secs			
Parameters per speaker	128	~ 25 million	512	512

Table 1: Comparison of speaker adaptation and speaker encoding approaches.



(a) (b)

Figure 4: (a) Speaker classification accuracy with different numbers of cloning samples. (b) EER (using 5 enrollment audios) for different numbers of cloning samples. LibriSpeech (unseen speakers) and VCTK represent EERs estimated from random pairing of utterances from ground-truth datasets.

source

. . .

VoiceLoop: Voice Fitting and Synthesis via A Phonological Loop

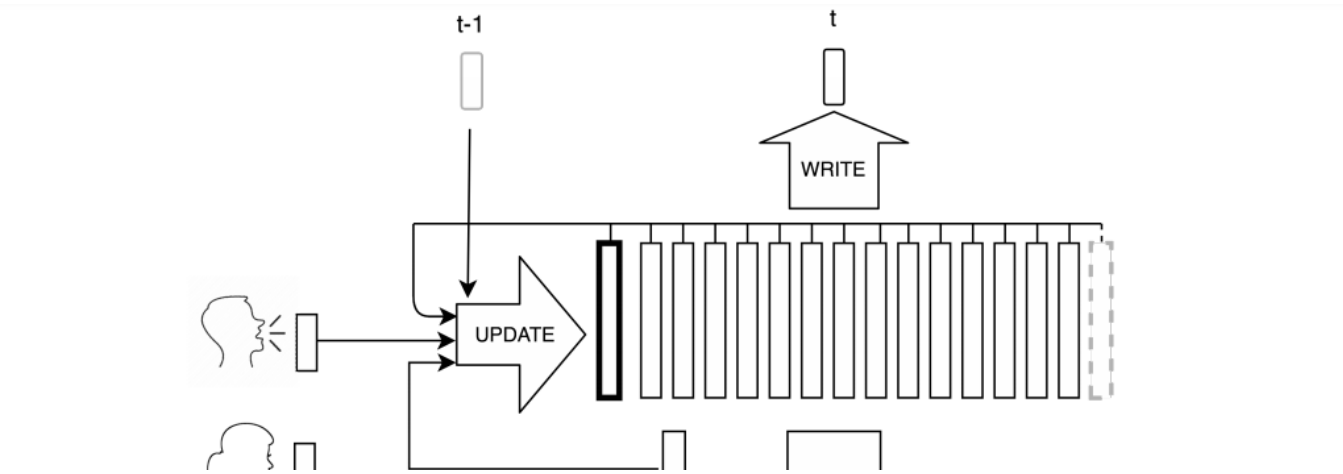
The authors of this paper are from Facebook AI Research. They introduce a neural text-to-speech (TTS) technique that can transform text into speech from voices that have been sampled from the wild.

> **VoiceLoop: Voice Fitting and Synthesis via a Phonological Loop**
>
> We present a new neural text to speech (TTS) method that is able to transform text to speech in voices that are sampled...
>
> arxiv.org

VoiceLoop is inspired by a working memory model known as a phonological loop, which holds verbal information for a short time. It's comprised of a phonological store that's constantly being replaced, and a rehearsal process that maintains longer-term representations in the phonological store.

VoiceLoop constructs a phonological store by implementing a shifting buffer as a matrix. Sentences are represented as a list of phonemes. A short vector is then decoded from each of the phonemes. The current context vector is generated by weighing the encoding of the phonemes and summing them at each time point.

Some of the properties that make VoiceLoop different include the use of a memory buffer instead of the conventional RNNs, memory sharing between all processes, and using shallow, fully-connected networks for all computations.



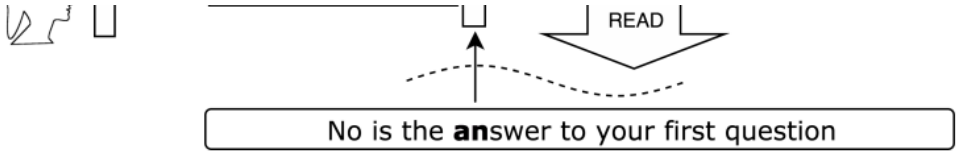


Figure 1: An overview of the VoiceLoop architecture. The reader combines the encoding of the sentence's phonemes using the attention weights to create the current context. A new representation is created by a shallow network that receives the context, the speaker ID, the previous output, and the buffer. The new representation is inserted into the buffer and the earliest vector in the buffer is discarded. The output is obtained by another shallow network that receives the buffer and the speaker as inputs. Once trained, fitting a new voice is done by freezing the network, except for the speaker embedding.

source

Below is a look at how the model performs in comparison to other alternatives.

Table 2: Single Speaker MOS Scores (Mean $\pm$ SD)

Method	LJ	Blizzard 2011	Blizzard 2013
Tacotron (re-impl)	2.06 ± 1.02	2.15 ± 1.10	N/A
Char2wav	3.42 ± 1.14	3.33 ± 1.06	2.03 ± 1.16
VoiceLoop	3.69 ± 1.04	3.38 ± 1.00	3.40 ± 1.03
Ground truth	4.60 ± 0.71	4.56 ± 0.67	4.80 ± 0.50

Table 3: Single Speaker MCD Scores (Mean $\pm$ SD; lower is better)

Method	LJ	Blizzard 2011	Blizzard 2013
Tacotron (re-impl)	12.82 ± 1.41	14.60 ± 7.02	N/A
Char2wav	19.41 ± 5.15	13.97 ± 4.93	18.72 ± 6.41
VoiceLoop	14.42 ± 1.39	8.86 ± 1.22	8.67 ± 1.26

Table 4: Multi-speaker MOS scores (Mean $\pm$ SE)

Method	VCTK22	VCTK65	VCTK85	VCTK101
Char2wav	2.84 ± 1.20	2.85 ± 1.19	2.76 ± 1.19	2.66 ± 1.16
VoiceLoop	3.57 ± 1.08	3.40 ± 1.00	3.13 ± 1.17	3.33 ± 1.10
GT	4.61 ± 0.75	4.59 ± 0.72	4.64 ± 0.64	4.63 ± 0.66

source

. . .

Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions

The authors of this paper are from Google and the University of California, Berkeley. They introduce Tacotron 2, a neural network architecture for speech synthesis from text.

> **Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions**
>
> This paper describes Tacotron 2, a neural network architecture for speech synthesis directly from text. The system is...
>
> arxiv.org

It's comprised of a recurrent sequence-to-sequence feature prediction network that maps character embeddings to mel-scale spectrograms. This is then followed by a WaveNet model that's been modified. This model acts as a vocoder that synthesizes time-domain waves from the spectrograms. The model achieves a mean opinion score (MOS) of 4.53.

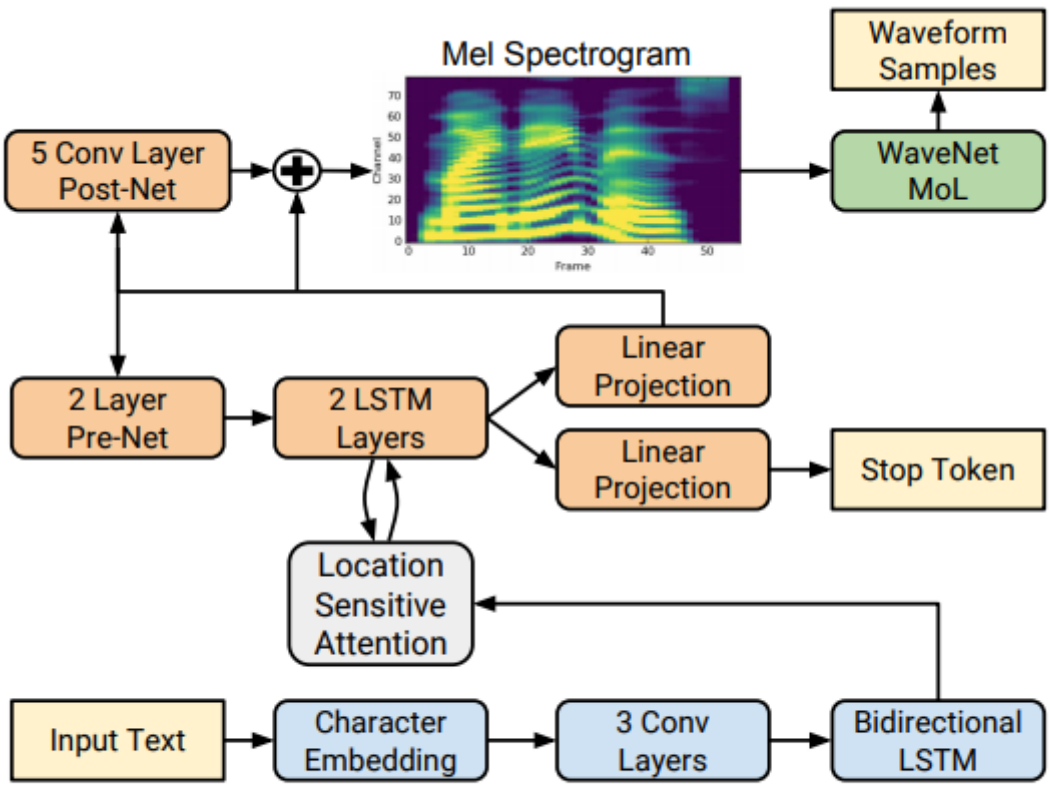


Fig. 1. Block diagram of the Tacotron 2 system architecture.

source

This model has been built by combining the best features of Tacotron and WaveNet. Below is the performance of the model in comparison to alternative models.

System	MOS
Parametric	3.492 ± 0.096
Tacotron (Griffin-Lim)	4.001 ± 0.087
Concatenative	4.166 ± 0.091
WaveNet (Linguistic)	4.341 ± 0.051
Ground truth	4.582 ± 0.053
Tacotron 2 (this paper)	$\mathbf{4.526 \pm 0.066}$

Table 1. Mean Opinion Score (MOS) evaluations with 95% confidence intervals computed from the t-distribution for various systems.

source

. . .

Conclusion

We should now be up to speed on some of the most common — and a couple of very recent — techniques for performing speech synthesis in a variety of contexts.

The papers/abstracts mentioned and linked to above also contain links to their code implementations. We'd be happy to see the results you obtain after testing them.

. . .

> Machine learning doesn't have to live on servers or in the cloud — it can also live on your smartphone. And Fritz has the tools to easily teach mobile apps to see, hear, sense, and think.

Editor's Note: Join Heartbeat on Slack and follow us on Twitter and LinkedIn for all the latest content, news, and more in machine learning, mobile development, and where the two intersect.

Thanks to Austin Kodra.